

Initiation au logiciel SAS

M1 MAS - Année universitaire 2023-2024

Alexie Robert

13 septembre 2023

Sommaire

- 1 Points clés de la séance du 8 septembre
- 2 Ce que l'on va aborder aujourd'hui

Les objectifs du TP précédent

- Créer et organiser son espace de travail ;
- lancer l'application SAS et se familiariser avec les différentes fenêtres ;
- créer des tables : étape DATA ;
- afficher les données : procédure PRINT ;
- créer des labels ;
- créer des formats ;
- afficher le dictionnaire : procédure CONTENTS ;
- utiliser l'instruction LIBNAME ;
- créer une macro-variable avec l'instruction %let.

Les objectifs du TP précédent

- Créer et organiser son espace de travail ;
- lancer l'application SAS et se familiariser avec les différentes fenêtres ;
- créer des tables : étape DATA ;
- afficher les données : procédure PRINT ;
- créer des labels ;
- créer des formats ;
- afficher le dictionnaire : procédure CONTENTS ;
- utiliser l'instruction LIBNAME ;
- créer une macro-variable avec l'instruction %let.

Des questions sur la correction du TP1 ?

L'étape DATA

Elle sert à

L'étape DATA

Elle sert à créer des tables SAS.

L'étape DATA

Elle sert à créer des tables SAS. Comment ?

L'étape DATA

Elle sert à créer des tables SAS. Comment ?

- 1 à partir d'un fichier de données (excel, txt,...) ;
- 2 en entrant directement des données ;
- 3 en partant d'une table existante.

L'étape DATA

Elle sert à créer des tables SAS. Comment ?

- 1 à partir d'un fichier de données (excel, txt,...) ;
 - 2 en entrant directement des données ;
 - 3 en partant d'une table existante.
-
- 1 A partir d'un fichier de données (excel, txt,...)

```
DATA insertion;  
  INFILE "C:\Insertion.txt";  
  LENGTH region $30;  
  INPUT region $ part txemp txcho salaire;  
  RUN;
```

L'étape DATA

- 2 en entrant directement des données

```
DATA minins ;
  LENGTH region $30 ;
  INPUT region $ part txemp txcho salaire ;
  CARDS;
Normandie 0.05 0.3523 0.2218 1400
Occitanie 0.084 0.4738 0.1697 1470
Corse 0.003 0.4015 0.1835 1500
; RUN;
```

L'étape DATA

3 en partant d'une table SAS existante

```
DATA insertion2 ;  
SET insertion ;  
LABEL  
region = "Région de formation"  
part = "Part de sortants"  
txemp = "Taux d'emploi"  
txcho = "Taux de chômage"  
salaire = "Salaire mensuel médian"  
;RUN;
```

Les bibliothèques

Temporaire ou permanente ?

Les bibliothèques

Temporaire ou permanente ?

```
/* Affectation de la bibliothèque permanente TP */
LIBNAME TP "C:\InitSAS";
```

Les bibliothèques

Temporaire ou permanente ?

```
/* Affectation de la bibliothèque permanente TP */
LIBNAME TP "C:\InitSAS";
```

```
/* Equivalent à */
%let chemin = C:\InitSAS ;
LIBNAME TP "&chemin";
```

Les labels et les formats

1 les labels

```
DATA insertion2 ;
  SET insertion ;
  LABEL
    region = "Région de formation"
    part = "Part de sortants"
    txemp = "Taux d'emploi"
    txcho = "Taux de chômage"
    salaire = "Salaire mensuel médian"
;RUN;
```

Les labels et les formats

2 les formats

```
PROC FORMAT;  
/* Pour une variable numérique */  
value clacho  
0-0.19999999 = "Faible"  
0.2-0.249999999999 = "Moyen"  
.25-1 = "Elevé";  
/* Pour une variable alphanumérique */  
value $ouinon  
"1" = "Oui"  
"2" = "Non";  
RUN;
```

La procédure print

1 le programme

```
/* Ajout d'un titre */
TITLE "L'insertion des jeunes selon leur région de formation";
/* Ajout d'une note de bas de page */
FOOTNOTE "Données provenant du Céreq";
PROC PRINT DATA=minins;
RUN;
/* Suppression du titre et de la note de bas de page
pour les sorties suivantes */
TITLE;
FOOTNOTE;
```

La procédure print

1 le programme

```
/* Ajout d'un titre */
TITLE "L'insertion des jeunes selon leur région de formation";
/* Ajout d'une note de bas de page */
FOOTNOTE "Données provenant du Céreq";
PROC PRINT DATA=minins;
RUN;
/* Suppression du titre et de la note de bas de page
pour les sorties suivantes */
TITLE;
FOOTNOTE;
```

2 la sortie

L'insertion des jeunes selon leur région de formation

Obs.	region	part	txemp	txcho	salaire
1	Normandie	0.050	0.3523	0.2218	1400
2	Occitanie	0.084	0.4738	0.1697	1470
3	Corse	0.003	0.4015	0.1835	1500

Données provenant du Céreq

La procédure contents

1 le programme

```
/* Affichage du dictionnaire */
PROC CONTENTS DATA=insertion;
RUN;
```

La procédure contents

1 le programme

```
/* Affichage du dictionnaire */  
PROC CONTENTS DATA=insertion;  
RUN;
```

2 la sortie

Liste alphabétique des variables et des attributs				
#	Variable	Type	Long.	Libellé
2	part	Num.	8	Part de sortants
1	region	Texte	30	Région de formation
5	salaire	Num.	8	Salaire mensuel médian
4	txcho	Num.	8	Taux de chômage
3	txemp	Num.	8	Taux d'emploi

Les objectifs du TP d'aujourd'hui

- savoir utiliser la procédure IMPORT ;
- créer de nouvelles variables dans une table SAS ;
- créer de nouvelles tables à partir d'une table SAS ;
- utiliser les procédures SORT et MERGE;
- découvrir les instructions DROP, KEEP et RETAIN.

L'import de données

Plusieurs options possibles :

L'import de données

Plusieurs options possibles :

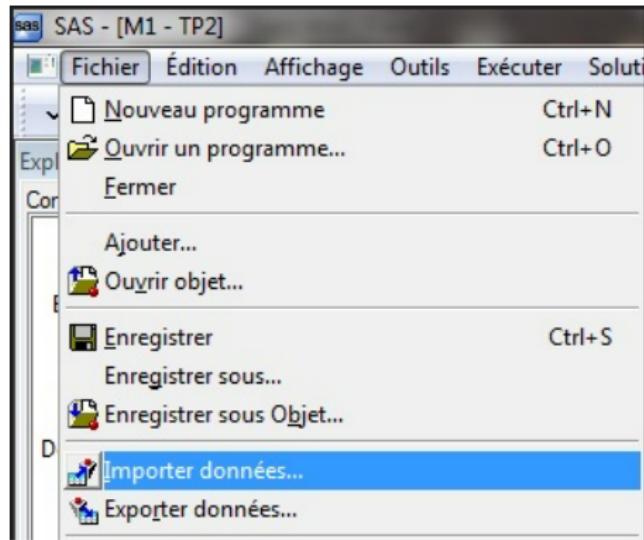
1. De manière manuelle

```
/* Import du fichier herp.txt */
/* Avec l'instruction INFILE */
data herp1;
INFILE "&path\herp.txt" FIRSTOBS=2 DELIMITER='09'x ;
LENGTH IDEN $5 SEXE $1 ;
INPUT IDEN $ SEXE $ AGE TAILLE POIDS
POULS SYS DIA CHOLNUM;
RUN;
```

```
/* Exemple d'import de données en format .csv avec séparateur virgule */
PROC IMPORT OUT= WORK.HERP
DATAFILE= "&path\herp.csv"
DBMS=CSV REPLACE;
GETNAMES=YES;
DATAROW=2;
RUN;
```

L'import de données

2. A l'aide de la boîte de dialogue



Création, manipulation de variables

```
DATA herp2;
  SET herp;
  TAILLEB=TAILLE/100;
  IMC=POIDS/ (TAILLEB*TAILLEB) ;
  if IMC>25 then SURPOIDS="1";
  if IMC<=25 then SURPOIDS="2";
RUN;
```

Création, manipulation de variables

```
□ DATA herp2;  
    SET herp;  
    TAILLEB=TAILLE/100;  
    IMC=POIDS/(TAILLEB*TAILLEB);  
    if IMC>25 then SURPOIDS="1";  
    if IMC<=25 then SURPOIDS="2";  
    RUN;  
  
□ DATA herp3;  
    SET herp2;  
    DROP SEXE;  
    RENAME SURPOIDS = SURP ;  
    RUN;  
  
□ PROC DELETE data=herp2;RUN;
```

Création de nouvelles variables

```
□ PROC FORMAT library=TP;  
  VALUE TRANCHE  
    0-19.9 = "Moins de 20 ans"  
    20-29.9 = "De 20 à 29 ans"  
    30-39.9 = "De 30 à 39 ans"  
    40-99.9 = "40 ans ou plus"  
  ;  
RUN;

---

  
options FMTSEARCH = (TP) ;  
□ DATA herp4;  
  SET herp3;  
  tage=put(age,tranche.);  
RUN;
```

Cas particulier : Gestion des dates

Une date SAS est un nombre entier égal au nombre de jours écoulés depuis le 1er janvier 1960. Les dates avant 1960 sont donc négatives.

Dans ce cas, on crée une nouvelle variable en appliquant un format de date.

Exemple : datenew=put(date,ddmmyy10.) qui contiendra des dates en format DD/MM/YYYY et permettra de faire des calculs.

Il existe d'autres formats de dates comme YEAR4. (2022), MMYYSD7. (09/2022), FRADFWDX. (9 septembre 2022).

Autre possibilité : Créer une date SAS avec la fonction MDY à partir d'une valeur numérique de mois, jour, et année :
MDY(var_mois,var_jour,var_an);

Cas particulier : Calculs à partir de dates

Fonctions utiles :

TODAY(); pour enregistrer la date d'aujourd'hui dans une variable.

MONTH(VAR1) Renvoie la valeur numérique du mois d'une variable en format date.

DAY(VAR1) Renvoie la valeur numérique du jour en format date.

YEAR(VAR1) Renvoie la valeur numérique de l'année en format date.

Soustraire deux dates renvoie l'écart entre deux dates en nombre de jours.

La fonction INTCK calcule l'intervalle de temps entre deux dates SAS.

L'intervalle peut être calculer en nombre d'année, mois ou jours.

Syntaxe : INTCK("month",DATE1,DATE2);

Création de tables

```
✉ data jeune vieux;  
      set herp;  
      if age<41 then output jeune;  
      else output vieux;  
run;
```

```
✉ data tous;  
      set jeune vieux;  
run;
```

Le tri des tables

```
/* Tri de la table musique par nombre de représentations croissant */
PROC SORT DATA=musique;
BY representations;
RUN;

/* Impression du contenu de la table musique dans la fenêtre sortie */
PROC PRINT DATA=musique;
RUN;
```

Le tri des tables

```
/* Tri de la table musique par nombre de représentations croissant */
PROC SORT DATA=musique;
BY representations;
RUN;

/* Impression du contenu de la table musique dans la fenêtre sortie */
PROC PRINT DATA=musique;
RUN;
```

Obs.	province	representations	art	recettes
1	Nouvelle-Ecosse	85	musique	2666410
2	Nouveau-Brunswick	128	musique	728585
3	Manitoba	130	musique	7547018
4	Saskatchewan	175	musique	2114121
5	Alberta	343	musique	14854751
6	Colombie-Britannique	534	musique	17703253
7	Ontario	1331	musique	57767233
8	Quebec	2035	musique	41967518

Le tri des tables

```
/* Tri de la table theatre par recettes décroissantes et
impression de la table theatre
dans la fenêtre sortie */
PROC SORT DATA=theatre;
BY DESCENDING recettes;
RUN;
PROC PRINT DATA=theatre;
RUN;
```

Le tri des tables

```
/* Tri de la table theatre par recettes décroissantes et
impression de la table theatre
dans la fenêtre sortie */
PROC SORT DATA=theatre;
BY DESCENDING recettes;
RUN;
PROC PRINT DATA=theatre;
RUN;
```

Obs.	province	representations	art	recettes
1	Ontario	11436	theatre	132930972
2	Quebec	9441	theatre	65615908
3	Colombie-Britannique	4868	theatre	28631452
4	Manitoba	2079	theatre	11483333
5	Ile du Prince Edouard	412	theatre	4490519
6	Saskatchewan	520	theatre	2935377
7	Nouveau-Brunswick	645	theatre	2587265
8	Terre-Neuve et Labrador	814	theatre	2155791
9	Yukon	20	theatre	327235

Suppression des doublons

La procédure SORT est également utilisée pour supprimer des lignes en double.

Exemple : Si on a plusieurs lignes par province, une par trimestre, on peut vouloir garder uniquement le trimestre avec la recette la plus élevée (différent selon la province) pour n'avoir qu'une ligne par province.

Programme correspondant :

```
/* Tri de la table theatrebis par provinces et par recettes décroissantes
au sein de celles-ci */
proc sort data=theatrebis;
  by province descending recettes;
run;

/*On ne garde dans la table theatre_ssdouble que la première ligne de
chaque province en repartant de la table triée précédemment */
proc sort nodupkey data=theatrebis out=theatre_ssdouble;
  by province;
run;
```

La fusion de tables

1 le programme

```
/* Tri des tables musique et theatre par nom de province croissant */
PROC SORT DATA=musique;BY province;RUN;
PROC SORT DATA=theatre;BY province;RUN;

/* Fusion des tables, tous les individus statistiques sont conservés */
DATA arts2a;
MERGE musique theatre;
BY province;RUN;
```

La fusion de tables

1 le programme

```
/* Tri des tables musique et theatre par nom de province croissant */
PROC SORT DATA=musique;BY province;RUN;
PROC SORT DATA=theatre;BY province;RUN;

/* Fusion des tables, tous les individus statistiques sont conservés */
DATA arts2a;
MERGE musique theatre;
BY province;RUN;
```

2 la sortie

Obs.	province	represmus	recetmus	represthea	recetthea
1	Alberta	343	14854751	-	-
2	Colombie-Britannique	534	17703253	4868	28631452
3	Ile du Prince Edouard	-	-	412	4490519
4	Manitoba	130	7547018	2079	11483333
5	Nouveau-Brunswick	128	728585	645	2587265
6	Nouvelle-Ecosse	85	2666410	-	-
7	Ontario	1331	57767233	11436	132930972
8	Quebec	2035	41967518	9441	65615908
9	Saskatchewan	175	2114121	520	2935377
10	Terre-Neuve et Labrador	-	-	814	2155791
11	Yukon	-	-	20	327235

La fusion de tables

1 le programme

```
/* Fusion des tables, seuls les individus statistiques  
présents dans les deux tables sont conservés */  
DATA arts3a;  
MERGE musique (IN=mu) theatre (IN=th);  
BY province;  
IF mu AND th;  
RUN;
```

La fusion de tables

1 le programme

```
/* Fusion des tables, seuls les individus statistiques
présents dans les deux tables sont conservés */
DATA arts3a;
MERGE musique (IN=mu) theatre (IN=th);
BY province;
IF mu AND th;
RUN;
```

2 la sortie

	province	represmus	recetmus	represthea	recethea
1	Colombie-Britannique	534	17703253	4868	28631452
2	Manitoba	130	7547018	2079	11483333
3	Nouveau-Brunswick	128	728585	645	2587265
4	Ontario	1331	57767233	11436	132930972
5	Quebec	2035	41967518	9441	65615908
6	Saskatchewan	175	2114121	520	2935377